Lab 3 (Team Lab): Robot

Start: After demo Lab 2

Demo Due: October 312014

Stand-Alone Program Demo and Code Explanation (*all team members must be present*)
your team will demonstrate your **Lab 3 assignment (both Part A and B).**

Your team may want to demo Part A first. Part B must be in a stand-along mode and use power from a 9-V battery (instead of USB).

Your team can get replacements of your 9-V battery at the ECE stockroom

During the demo, your team will need to both provide a demonstration of your working stand-alone programmed board and provide an explanation of your code.

Points: 50

C Code Submission on D2L: All C files for each lab assignment and final project should be submitted on D2L in the Dropbox. After your demo, ***submit your C file*** on D2L Dropbox by ***11.59 PM on October 31st***.

Make sure you carefully read this entire handout.

Part A in this lab should help you with Part B (controlling the motors of the robot)

Lab Overview: In this lab,

- In this lab, you will build a DC motor controller that utilizes a **potentiometer to control the speed of two DC motors** and a **momentary switch to control their direction.**

  Analog to Digital Conversion (ADC) must be utilized to determine the potentiometer position and **Pulse width modulation** (i.e. output compare module) must be utilized to control the speed of the motors. No credit will be awarded if loops and delays are utilized for controlling the motor speed.

- In the first part (Part A), you will build the system with the LCD, ADC and PWM and measure/test the PWM signals using the oscilloscope and display required value on LCD.

Wire Wrapping and Color Coding Requirements:
The provided jumper wires are free to be used for developing and testing your system, however, when it comes to the ***final demo, everything should be* wire wrapped/soldered *cleanly with "reasonable" color coding*** (using black wires all the time is obviously a BAD color coding), otherwise 5% of the points will be deducted.

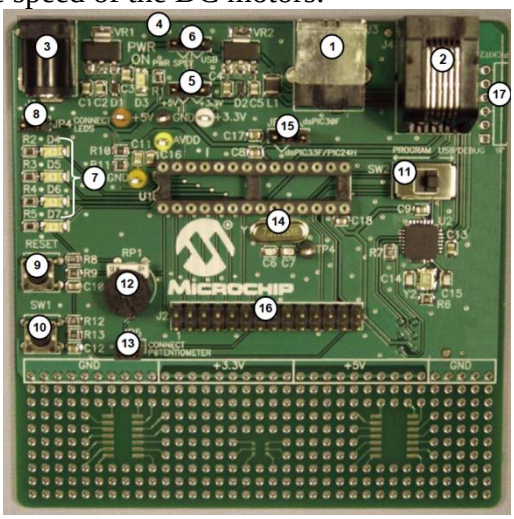
Datasheets and References (on D2L)
H-Bridge L293B Datasheet
LCD Display Datasheet
Microchip 16-bit 28-pin Starter Board User's Guide and PICkit 3 User's Guide
PIC24FJ64GA002 Datasheet and PIC24F Family Reference Manuals

Provided Source Code: No Source Code *(you're on your own this time)*

Parts Needed for Lab (all parts are provided in your team's Parts Kit):
1 - LCD
1 - H-Bridge L293B IC chip
1 – Robot/Motor Kit [https://www.sparkfun.com/products/10825]
1 - Power Connector (Female 2.5mm)
Male Headers

Lab Part A: ADC, PWM, with LCD

1. Use ***Analog to Digital Conversion (ADC)*** of the microcontroller to read the (voltage) **value of the potentiometer** *(labeled 12 in the following picture)* on the 16-bit 28-pin Starter Board and display its digital value on the 1st line of the LCD. In other words, ADC is used to determine the potentiometer position.
   In Part B, the potentiometer position (together with PWM signals (Part 2 below)) will be utilized to control the speed of the DC motors.



2. Use the ***output compare module*** of the microcontroller to create ***two PWM signals***. The duty cycle for each PWM signal should be controlled by the potentiometer position as described below. In part B, each PWM signal will be used to controller the speed of each motor.
   - Use the oscilloscope to measure/display the PWM signals that your system generates.
   - Display the duty cycle for each PWM signal on the 2nd line of the LCD.

   a) If the potentiometer is in the middle position, both motors should operate at maximum speed.
   b) If the potentiometer is turned fully to the right, the left motor should operate at full speed, and the right motor should be stopped *(this will achieve maximum turning in the right direction)*.
   c) If the potentiometer is turned fully to the left, the right motor should operate at full speed, and the left motor should be stopped *(this will achieve maximum turning in the left direction)*.
   d) ***All potentiometer positions between these settings should correspond to fractional values for the direction and control of the motors.*** In other words, your software should provide a smooth, continuous transition between all operating directions.

Lab Part B: Motor Control with Robot

NOTE: *Before applying 9V-battery power to your circuit, double check all your connections or ask TA for help to ensure that you have properly connected the circuit i.e. power pins, H-bridge, ...*

Hardware Connection:

- Build your team's car/robot from a robot kit
- Connect the H-bridge IC chip to your microcontroller and the motors of the car/robot
- Both the PIC board and the DC motors should be powered by the 9V battery pack (you should do this step after you are sure that your code functions correctly with the h-bridge chip and dc motors).

Software:

1. Continue using LCD to display required values as stated in part A.
2. The ***potentiometer*** on the 16-bit 28-pin Starter Board should be utilized to control the speed of the DC motors. ***Analog to Digital Conversion (ADC)*** must be utilized to determine the potentiometer position.
    - If the potentiometer is the middle, both motors should operate at maximum speed in the current direction.
    - If the potentiometer is turned fully to the right, the left motor should operate at full speed, and the right motor should be stopped *(this will achieve maximum turning in the right direction)*.
    - If the potentiometer is turned fully to the left, the right motor should operate at full speed, and the left motor should be stopped *(this will achieve maximum turning in the left direction)*.
    - All potentiometer positions between these settings should correspond to fractional values for the direction and control of the motors. In other words, your software should provide a smooth, continuous transition between all operating directions.
3. A ***momentary switch*** should be utilized to switch between three operating states, *Idle*, *Forward*, *Backwards*.
    1. For every distinct button press, the operating states should cycle through the sequence, *Idle*, *Forward*, *Idle*, *Backwards (repeat)*.
    2. In *Idle* mode, both motors should be stopped.
    3. In *Forward* mode, the motors should move in a forward direction subject to the direction control defined above.
    4. In *Backward* mode, the motors should move in a backward direction (i.e. opposite direction of Forward mode) subject to the direction control defined above.
4. ***Pulse width modulation*** (i.e. output compare module) must be utilized to ***control the speed of the motors***. No lab point/credit will be awarded if loops and delays are utilized for controlling the motor speed.
5. The ***H-Bridge*** IC chip must be utilized to ***interface with the DC motors***.
6. Both the PIC board and the DC motors (robot) should be powered by the provided 9V battery pack, i.e. ***NO*** external power supply, ***NO*** USB cable connected to the board. The PIC board should be configured to run in the ***stand alone*** mode.

Once your car/robot starts moving, your team may have to adjust the code (PWM for left and right motors) in order for the robot to be able to move straight due to the imperfection of the motors/tires on either side.

C Coding and Commenting Style Guidelines

a. All files should have the names and date. If working in a group, then all group members should be listed.
b. All functions should have a descriptive note of what it does.
c. All variables should have a descriptive name and a comment on what it's used for.
d. Any PIC configuration setting should have a comment indicating what the configuration corresponds to.
e. In the later labs, if you can use an interrupt instead of a busy wait, then you should use the interrupt.

Note: Class examples don't always adhere to this standard, be more diligent then we were.